

# NDD(Near Duplicate Detector) 使用方法

2009/03/27

## 1 概要

### 1.1 準同一映像区間検出アルゴリズム

準同一映像区間とは、映像中に繰り返し現れる CM やテロップは異なるものの同じ素材映像を用いたニュース映像などの映像区間を指します。従来、長時間映像中からこれらの準同一映像区間をもれなく検出する処理は、膨大な計算時間を要することから非常に困難でしたが、本アルゴリズムは、この処理を高速に行うものです。具体的には、空間方向と時間方向の 2 段階の次元圧縮と低次元と高次元の 2 段階の照合により、この高速化を実現しています。

### 1.2 準同一映像区間検出ソフトウェア: NDD (Near Duplicate Detector)

NDD は上記のアルゴリズムを実装した LINUX 上のコマンドラインで動作する C 言語のプログラムです。NDD の実行には、次元圧縮のための圧縮基底を学習するための学習用映像と準同一映像区間検出の対象となる検出用映像を mpeg ファイル形式で用意する必要があります。NDD は以下の 6 つのプログラムから構成されています。

- mkfvd: フレームベクトルの次元圧縮
- mkffs: フレームベクトルの圧縮基底の作成
- mkffd: フレームベクトルの次元圧縮
- mkfsfs: 映像区間ベクトルの圧縮基底の作成
- mkfsfd: 映像区間ベクトルの次元圧縮
- search: 準同一映像区間の検出

以下でこれらを説明します。

#### 1.2.1 mkfvd: 映像からのフレームベクトル抽出

mpeg ファイルを入力として、各フレーム画像に対して画像内のピクセル輝度値を要素として持つベクトルをフレームベクトルとして抽出し、フレームベクトルをフレーム数分並べた fvd ファイルを出力します。フレーム画像内で使用する領域を指定する機能と、ブロックサイズを指定してブロック内のピクセル輝度値を平均することによってフレーム画像の低解像度化を行う機能が用意されています。

### 1.2.2 mkffs: フレームベクトルの圧縮基底の作成

fvd ファイルを入力として、ファイル内のフレームベクトルを主成分分析することにより得られる固有ベクトルを固有値の高い順番に並べたものを、フレームベクトルの圧縮基底とし、ffs ファイルを出力します。圧縮に使用する固有ベクトルの数を直接、あるいは累積寄与率の形で指定することが可能です。

### 1.2.3 mkffd: フレームベクトルの次元圧縮

fvd ファイルと ffs ファイルを入力として、ffd ファイル内のフレームベクトルを、ffs ファイルの圧縮基底により次元圧縮した低次元のフレームベクトルを並べた ffd ファイルを出力します。

### 1.2.4 mkdfs: 映像区間ベクトルの圧縮基底の作成

ffd ファイルと区間長を入力として、ファイル内のフレームベクトルを指定された区間長分並べたベクトルを映像区間ベクトルとし、それらを主成分分析することにより得られる固有ベクトルを固有値の高い順番に並べたものを、映像区間ベクトルの圧縮基底とし、sfs ファイルを出力します。圧縮に使用する固有ベクトルの数を直接、あるいは累積寄与率の形で指定することが可能です。また、準同一映像検出は、ここで指定された区間長を基準として行われます。

### 1.2.5 mkdfd: 映像区間ベクトルの次元圧縮

ffd ファイルと sfs ファイルを入力とし、ffd ファイル内のフレームベクトルを mkdfs 時に指定された区間長分並べた映像区間ベクトルを、sfs ファイルの圧縮基底により次元圧縮した低次元の映像区間ベクトルを並べた sfd ファイルを出力します。

### 1.2.6 search: 準同一映像区間の検出

sfd ファイル、ffd ファイル、fvd ファイル、準同一映像区間検出のためのしきい値を入力として、準同一映像区間を検出し、検出結果を標準出力します。しきい値はベクトル間のユークリッド距離で与え、ベクトル間の距離がしきい値以下のすべての映像区間対を準同一映像区間対として検出します。検出結果は、準同一映像区間対として検出されたベクトル間の距離、それぞれの区間の始点と終点の時刻として出力されます。

## 1.3 NDD の処理の流れ

上記の 6 つのプログラムを用いて、学習用と検出用の 2 つの mpeg ファイルから検出結果を出力するまでの流れを図 1 に示します。NDD の処理は、圧縮基底を学習する段階と準同一映像を検出する段階の 2 つの段階に分かれます。

学習段階では、学習用 mpeg ファイルを入力として、フレームベクトルの圧縮基底である ffs ファイルと映像区間ベクトルの圧縮基底である sfs ファイルを作成します。学習用映像の選び方によって、後に続く検出段階に要する処理時間に変化が生じます。学習用映像が検出用映像と類似した性質の映像である場合には検出処理の時間が短くなり、逆の場合には長くなります。そのため、検出

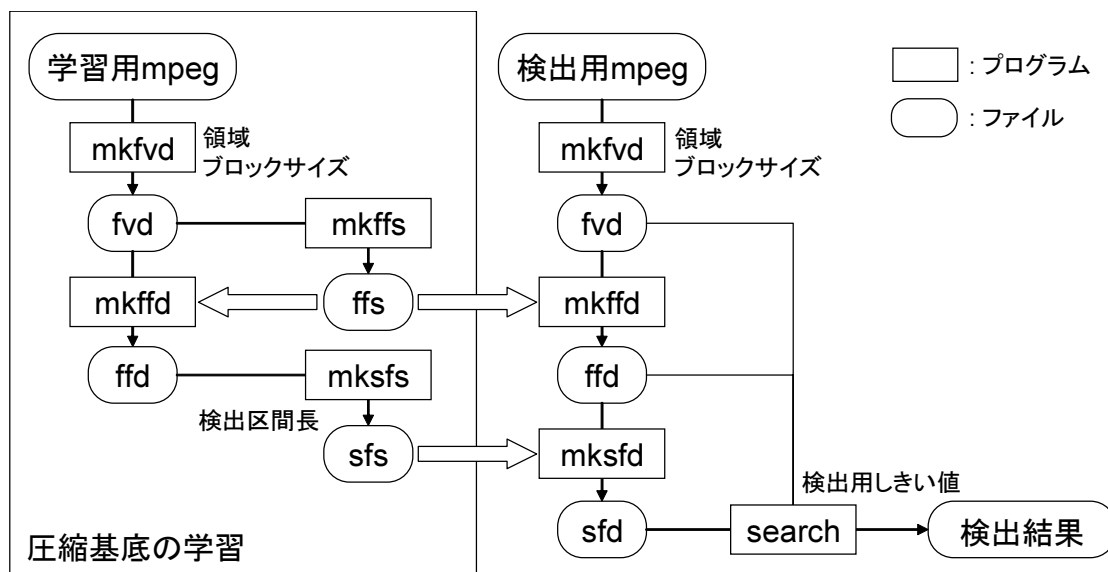


図 1: NDD の処理の流れ

用映像の性質が事前に分かっている場合にはそれと類似した学習用映像を用い、分からない場合にはなるべく様々な性質の映像を含む学習用映像を用いることが好ましいです。

検出段階では、初めに検出用 mpeg ファイルから fvd ファイルを作成します。この際、フレーム画像内の検出に使用する領域と低解像度化のためのブロックサイズとして、学習段階で指定したものと同一ものを指定する必要があります。次に fvd ファイルと学習段階で作成された ffs ファイルから ffd ファイルを、ffd ファイルと学習段階で作成された sfs ファイルから sfd をそれぞれ作成します。ここで、検出対象となる区間長は学習段階で sfs ファイルを作成した際に指定したものとなります。最後にこれまでに得られた fvd ファイル、ffd ファイル、sfd ファイルを入力として、準同一区間検出のしきい値を指定することにより、検出結果を得ます。mkfvd で指定した領域、ブロックサイズ、並びに mksfs で指定した検出対象区間長を変える必要が無い場合には、学習段階で得られた ffs ファイルと sfs ファイルをそのまま利用して、別の検出用 mpeg ファイルからの準同一区間検出処理を行うことができます。

## 2 プログラムが生成するファイル

いずれもバイナリファイルで、下記の 5 種類のファイルを生成します。

ファイル種類	拡張子
frame vector ファイル	fvd
frame feature selector ファイル	ffs
frame feature ファイル	ffd
segment feature selector ファイル	sfs
segment feature ファイル	sfd

## 3 プログラムの使用方法

### 3.1 mkfvd : MPEG ファイルから frame vector を抽出する

書式

```
% mkfvd --mpeg=<InputMpegFile>
        --fvd=<OutputFvdFile>
        [--region=<RegionRectangle>]
        [--block=<BlockSize>]
        [--color-resolution=<ColorResolution>]
```

説明 MPEG ファイルを入力として、フレームデータの取得、色の量子化・正規化、などの処理を行って、fvd ファイルを出力します。

InputMpegFile には、入力となる MPEG ファイルのファイル名を指定します。必ず指定しなければなりません。

OutputFvdFile には、出力となる fvd ファイルのファイル名を指定します。必ず指定しなければなりません。

RegionRectangle には、入力ファイルから得られるフレームデータのうち、どの領域を処理に使用するかを、, (カンマ) で区切ったピクセル単位の座標で指定します。指定は任意です。

指定の仕方は、順に左上の X 座標、同 Y 座標、幅、高さです。下記の例では、左上を基準として (10, 20) – (210, 120) の矩形が処理の対象となります。

```
--region=10,20,200,100
```

このオプションを指定しなかった場合は、--region=0,0,0,0 が与えられたものとして処理します。

BlockSize には、入力ファイルから得られるフレームデータをブロック単位で正規化する際のブロックサイズを、, (カンマ) で区切ったピクセル単位の幅と高さで指定します。指定は任意です。

指定の仕方は、順に幅、高さです。下記の例では、縦 16 ピクセル、横 8 ピクセルの単位で正規化を行います。

```
--block=8,16
```

このオプションを指定しなかった場合は、--block=16,16 が与えられたものとして処理します。

ColorResolution には、入力ファイルから得られるフレームデータのピクセル値を量子化する精度を指定します。指定は任意です。

このオプションを指定しなかった場合は、--color-resolution=256 が与えられたものとして処理します。

使用例

```
% mkfvd --mpeg=VIDEO.mpg --fvd=hoge.fvd \
        --region=0,0,352,240 \
        --block=16,16
```

### 3.2 mkffs : fvd ファイルから、frame feature selector を計算する

書式

```
% mkffs --fvd=<InputFvdFile>
        --ffs=<OutputFfsFile>
        [--sample-count=<SampleCount>]
        [--dimension=<Dimension> | --proportion=<Proportion>]
```

説明 fvd ファイルを入力として、フレームデータを元に主成分分析を行い、次元圧縮のための係数ベクトルデータからなる ffs ファイルを出力します。

--fvd=<InputFvdFile>には、入力となる fvd ファイルのファイル名を指定します。必ず指定しなければなりません。

--ffs=<OutputFfsFile>には、出力となる ffs ファイルのファイル名を指定します。必ず指定しなければなりません。

--sample-count=<SampleCount>には、入力ファイルのうち何フレーム分のデータを使用して主成分分析を行うかを指定します。指定は任意で、省略した場合は--sample-count=15000 が与えられたものとして処理します。

--dimension=<Dimension>には、主成分分析の係数ベクトルを何次元分出力するかを指定します。

--proportion=<Proportion>には、主成分分析の係数ベクトルを何次元分出力するかを指定しますが、次元数を直接指定するのではなく、累積寄与率の上限を指定します。

--dimension および --proportion はどちらか一方を指定できます。また、省略することも可能で、省略した場合は--dimension=10 が与えられたものとして処理します。

使用例

```
% mkffs --fvd=hoge.fvd --ffs=hoge.ffs \
        --sample-count=15000 --dimension=10
```

### 3.3 mkffd : fvd ファイルと ffs ファイルから、frame feature を計算する

書式

```
% mkffd --fvd=<InputFvdFile>
        --ffs=<InputFfsFile>
        --ffd=<OutputFfdFile>
```

説明 fvd ファイルを入力とし、ffs ファイルから得られるベクトルを用いて次元圧縮を行い、結果を ffd ファイルとして出力します。

--fvd=<InputFvdFile>には、入力となる fvd ファイルのファイル名を指定します。必ず指定しなければなりません。

--ffs=<InputFfsFile>には、入力となる ffs ファイルのファイル名を指定します。必ず指定しなければなりません。

--ffd=<OutputFfdFile>には、出力となる ffd ファイルのファイル名を指定します。必ず指定しなければなりません。

## 使用例

```
% mkffd --fvd=hoge.fvd --ffs=hoge.ffi --ffd=hoge.ffd
```

### 3.4 mksfs : ffd ファイルから、segment feature selector を計算する

#### 書式

```
% mksfs --ffd=<InputFfdFile>
      --sfs=<OutputSfsFile>
      [--size=<SegmentSize>]
      [--sample-count=<SampleCount>]
      [--dimension=<Dimension> | --proportion=<Proportion>]
```

説明 ffd ファイルを入力として、フレームを結合したベクトルを元に主成分分析を行い、次元圧縮のための係数ベクトルデータからなる sfs ファイルを出力します。

--ffd=<InputFfdFile>には、入力となる ffd ファイルのファイル名を指定します。必ず指定しなければなりません。

--sfs=<OutputSfsFile>には、出力となる sfs ファイルのファイル名を指定します。必ず指定しなければなりません。

--size=<SegmentSize>には、いくつかのフレームを結合したもとを使用して後の処理を行うかを指定します。指定は任意で、省略した場合は--size=150 が与えられたものとして処理します。

--sample-count=<SampleCount>には、入力ファイルのうち、フレームを結合したデータをいくつ使用して主成分分析を行うかを指定します。指定は任意で、省略した場合は--sample-count=15000 が与えられたものとして処理します。

--dimension=<Dimension>には、主成分分析の係数ベクトルを何次元分出力するかを指定します。

--proportion=<Proportion>には、主成分分析の係数ベクトルを何次元分出力するかを指定しますが、次元数を直接指定するのではなく、累積寄与率の上限を指定します。

--dimension および --proportion はどちらか一方を指定できます。また、省略することも可能で、省略した場合は--dimension=10 が与えられたものとして処理します。

## 使用例

```
% mksfs --ffd=hoge.ffd --sfs=hoge.sfs \
      --size=150 \
      --sample-count=15000 --dimension=10
```

### 3.5 mkafd : ffd ファイルと sfs ファイルから、segment feature を計算する

#### 書式

```
% mkafd --ffd=<InputFfdFile>
      --sfs=<InputSfsFile>
      --afd=<OutputSfdFile>
```

説明 ffd ファイルを入力とし、sfs ファイルから得られるベクトルを用いて次元圧縮を行い、結果を sfd ファイルとして出力します。

--ffd=<InputFfdFile>には、入力となる ffd ファイルのファイル名を指定します。必ず指定しなければなりません。

--sfs=<InputSfsFile>には、入力となる sfs ファイルのファイル名を指定します。必ず指定しなければなりません。

--sfd=<OutputSfdFile>には、出力となる sfd ファイルのファイル名を指定します。必ず指定しなければなりません。

#### 使用例

```
% mksfd --ffd=hoge.ffd --sfs=hoge.sfs --sfd=hoge.sfd
```

### 3.6 search : 1 組の sfd ファイル、ffd ファイルと fvd ファイルから、一致箇所を抽出する

#### 書式

```
% search --sfd=<InputSfdFile>  
        --ffd=<InputFfdFile>  
        --fvd=<InputFvdFile>  
        [--sfd-threshold=<Distance>]  
        [--ffd-threshold=<Distance>]  
        [--fvd-threshold=<Distance>]  
        [--minimum-segment-size<FrameCount>]
```

説明 sfd ファイル、ffd ファイル、fvd ファイルを入力として、そのファイルの内部で類似する区間を検出し、出力します。

--sfd=<InputSfdFile>には、入力となる sfd ファイルのファイル名を指定します。必ず指定しなければなりません。

--ffd=<InputFfdFile>には、入力となる ffd ファイルのファイル名を指定します。必ず指定しなければなりません。

--fvd=<InputFvdFile>には、入力となる fvd ファイルのファイル名を指定します。必ず指定しなければなりません。

--sfd=<InputSfdFile>で指定するファイルは、--ffd=<InputFfdFile>で指定するファイルを元に作成されたものでなければなりません。

--ffd=<InputFfdFile>で指定するファイルは、--fvd=<InputFvdFile>で指定するファイルを元に作成されたものでなければなりません。

--minimum-segment-size=<FrameCount>には、抽出する一致区間の最小の長さをフレーム数で指定します。

#### 使用例

```
% search --sfd=hoge.sfd --ffd=hoge.ffd --fvd=hoge.fvd
```

### 3.7 search2 : 2組の sfd ファイル、ffd ファイルと fvd ファイルから、一致箇所を抽出する

#### 書式

```
% search --sfd1=<InputSfdFile>
        --ffd1=<InputFfdFile>
        --fvd1=<InputFvdFile>
        --sfd2=<InputSfdFile>
        --ffd2=<InputFfdFile>
        --fvd2=<InputFvdFile>
        [--sfd-threshold=<Distance>]
        [--ffd-threshold=<Distance>]
        [--fvd-threshold=<Distance>]
        [--minimum-segment-size<FrameCount>]
```

説明 sfd ファイル、ffd ファイル、fvd ファイルを入力として、そのファイルの内部で類似する区間を検出し、出力します。

--sfd1=<InputSfdFile>と--sfd2=<InputSfdFile>には、入力となる sfd ファイルのファイル名を指定します。必ず指定しなければなりません。

--ffd1=<InputFfdFile>と--ffd2=<InputFfdFile>には、入力となる ffd ファイルのファイル名を指定します。必ず指定しなければなりません。

--fvd1=<InputFvdFile>と--fvd2=<InputFvdFile>には、入力となる fvd ファイルのファイル名を指定します。必ず指定しなければなりません。

--sfd1=<InputSfdFile>で指定するファイルは、--ffd1=<InputFfdFile>で指定するファイルを元に作成されたものでなければなりません。

--ffd1=<InputFfdFile>で指定するファイルは、--fvd1=<InputFvdFile>で指定するファイルを元に作成されたものでなければなりません。

--sfd2=<InputSfdFile>で指定するファイルは、--ffd2=<InputFfdFile>で指定するファイルを元に作成されたものでなければなりません。

--ffd2=<InputFfdFile>で指定するファイルは、--fvd2=<InputFvdFile>で指定するファイルを元に作成されたものでなければなりません。

--minimum-segment-size=<FrameCount>には、抽出する一致区間の最小の長さをフレーム数で指定します。

#### 使用例

```
% search --sfd1=hoge.sfd --ffd1=hoge.ffd --fvd1=hoge.fvd \
        --sfd2=fuga.sfd --ffd2=fuga.ffd --fvd2=fuga.fvd
```